ON GRADIENT METHODS FOR APPROACHING CONSTRAINED MAXIMA

Kenneth J. Arrow
Leonid Hurwicz

APPENDIX by Thomas Marschak

P-1015

1 February 1957

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA

CLEARINGHOUSE FOR FEDERAL SCIENTIFIC AND TECHNICAL INFORMATION, CFSTI
DOCUMENT MANAGEMENT BRANCH 410.11

LIMITATIONS IN REPRODUCTION QUALITY

Accession # AD 606 264

☒ 1. We regret that legibility of this document is in part unsatisfactory. Reproduction has been made from best available copy.

☐ 2. A portion of the original document contains fine detail which may make reading of photocopy difficult.

☐ 3. The original document contains color, but distribution copies are available in black-and-white reproduction only.

☐ 4. The initial distribution copies contain color which will be shown in black-and-white when it is necessary to reprint.

☐ 5. Limited supply on hand; when exhausted, document will be available in Microfiche only.

☐ 6. Limited supply on hand; when exhausted document will not be available.

☐ 7. Document is available in Microfiche only.

☐ 8. Document available on loan from CFSTI (TT documents only).

☐ 9.

Processor: [signature]

NBS____ 9/64

ON GRADIENT METHODS FOR APPROACHING CONSTRAINED MAXIMA[1]

by

Kenneth J. Arrow
Leonid Hurwicz

1. A Gradient Method for Unconstrained Maxima.

Let f be a real-valued function of the n-dimensional variable $x = (x_1,\ldots,x_n)$. The n-dimensional Euclidean space of the x's will be denoted by X. Suppose $\hat{x}$ maximizes[2] f over the space X i.e.,

(1) $f(\hat{x}) \geq f(x)$ for all x in X.

Suppose f is analytic and strictly concave.[3] Then it is known[4] that the system of differential equations,

(2) $dx_i/dt = \partial f/\partial x_i$, $i = 1,\ldots,n$,

has a set of solutions,

(3) $x_i = X_i\,[t; X(0)]$, $i = 1,\ldots,n$,

converging to the maximum. I.e.,

(4) $\lim_{t\to\infty} X_i\,[t; X(0)] = \hat{x}_i$, $i = 1,\ldots,n$.

Because the vector $f_x = (\partial f/\partial x_1,\ldots,\partial f/\partial x_n)$ is known as the gradient of the function f, the process defined by (2) is called the gradient process.[5]

---

1. This is a more technical exposition of the ideas sketched in [1].

2. Minimization of g is equivalent to maximization of $-g$.

3. The function f is said to be (strictly) concave if for any $x^1 \neq x^2$ and any θ, $0 < \theta < 1$, $f[\theta x^1 + (1-\theta) x^2] \geq (>)\ \theta f(x^1) + (1-\theta) f(x^2)$.

4. See [2], pp. 301-2.

5. When the gradient method is applied to a minimization problem, it is sometimes called the method of steepest descent.

Its interest in the present context is due to the fact that it can be used as a computational technique for solving extremization problems. This is done by constructing finite difference approximations to (2) and carrying on the process until the x_i's appear to be undergoing little change.

2. Other Kinds of Extrema.

It is natural to consider two types of extensions of the gradient process. On the one hand, one might wish to apply it to a "stationary" point other than an extremum, viz., to the saddle-point of a function. On the other hand, one may use it in connection with extrema subject to constraints. In the light of the Kuhn-Tucker theorem cited in section 4 below, the two extensions are closely related.

3. A Gradient Method for Saddle-Points.

Let $\mathcal{U}$ and $\mathcal{V}$ be Euclidean spaces of dimensionalities p and q, respectively, and let $U \subseteq \mathcal{U}$, $V \subseteq \mathcal{V}$ be some given fixed subsets.[1] A real-valued function $\psi(u, v)$, $u \in U$, $v \in V$, is said to have a saddle-point over U, V at $(\bar{u}, \bar{v})$ if $\bar{u} \in U$, $\bar{v} \in V$,

and,

$$\psi(u, \bar{v}) \leq \psi(\bar{u}, \bar{v}) \leq \psi(\bar{u}, v)$$

for all $u \in U$, $v \in V$. (Any unconstrained maximization (resp. minimization) problem may be viewed as a special case of such a saddle-point with $q = 0$ and $U = \mathcal{U}$ (resp. $p = 0$ and $V = \mathcal{V}$).)

Since the saddle-point $(\bar{u}, \bar{v})$ is a maximum in u and a minimum in v, it is natural to devise a gradient process of the form,

---

1. In subsequent application, U and V are convex cones with vertices at the origin.

$$du_i/dt = \partial\Psi/\partial u_i, \qquad dv_j/dt = -\partial\Psi/\partial v_j,$$

(with additional rules for points on the boundaries of U and V).

Consider the case where, for some fixed $P \subseteq \{1,\ldots,p\}$, $Q' \subseteq \{1,\ldots,q\}$,

$$U = \{(u_1,\ldots,u_p): u_i \geq 0,\ i \in P'\},$$

$$V = \{(v_1,\ldots,v_q): v_j \geq 0,\ j \in Q'\},$$

define the gradient process by the differential equations,

$$(5) \qquad du_i/dt = \begin{cases} 0 \text{ if } u_i = 0,\ \partial\Psi/\partial u_i < 0 \text{ and } i \in P', \\ \partial\Psi/\partial u_i \text{ otherwise,} \end{cases}$$

$$dv_j/dt = \begin{cases} 0 \text{ if } v_j = 0,\ \partial\Psi/\partial v_j > 0 \text{ and } j \in Q', \\ -\partial\Psi/\partial v_j \text{ otherwise,} \end{cases}$$

and write the solutions of (5) as,

$$u_i[t; u(0), v(0)], \qquad v_j[t; u(0), v(0)].$$

Under what conditions do we have convergence in u_i (i.e., $\lim_{t\to\infty} u_i(t) = \bar{u}_i$, $i = 1,\ldots, p$)? A partial answer was given in [3]. To state the result, note that, at the saddle-point $(\bar{u}, \bar{v})$,

$$\partial\Psi/\partial u_i \leq 0 \text{ for each } i = 1,\ldots, p$$

$$\partial\Psi/\partial u_i = 0 \text{ for each } i \notin P',$$

and

$$(6) \qquad \bar{u}_i = 0 \text{ if } \partial\Psi/\partial u_i < 0 \text{ at } (\bar{u}, \bar{v}).$$

If (6) holds for $i = i_o$, we shall call i_o a <u>corner index</u>, otherwise an <u>interior index</u>. We write $u = (u^{(1)}, u^{(2)})$, where $u^{(1)}$ consists of corner indices, $u^{(2)}$ of interior indices. Analogously, j is a corner index for v if $\bar{v}_j = 0$ and $\partial\Psi/\partial v_j < 0$ at $(\bar{u}, \bar{v})$. Theorem 1 in [3] then states the following:

Let $\Psi(u, v)$ be linear[1] in y, possess a saddle-point at $(\bar{u}, \bar{v})$ over[2]

$$U = \left\{(u_1, \ldots, u_p);\ u_i \geq 0,\ i = 1, \ldots, p\right\},$$

$$V = \left\{(v_1, \ldots, v_q);\ v_j \geq 0,\ j = 1, \ldots, q\right\},$$

and be analytic in some neighborhood of $(\bar{u}, \bar{v})$. Suppose further that,[3]

(a) $\| \partial^2 \Psi / \partial u_{h'}^{(2)} \partial u_{h''}^{(2)} \|$ is negative definite at $(\bar{u}, \bar{v})$,

(b) $\bar{u}_i > 0$, $\bar{v}_j > 0$ for every interior index i or j.

Then for every initial position (u^0, v^0) in a sufficiently small neighborhood of $(\bar{u}, \bar{v})$, there is a unique solution $u_i(t; u^0, v^0)$, $v_j(t; u^0, v^0)$ of the gradient equations such that,

$$\lim_{t \to \infty} u_i(t; u^0, v^0) = \bar{u}_i,\ i = 1, \ldots, p.$$

This result was extended by H. Uzawa in 1955 in an unpublished note; he showed that the assertion of the preceding theorem is also valid in the large (and not only for (u^0, v^0) close to $(\bar{u}, \bar{v})$) when Ψ is assumed strictly concave in u for each y.[4]

4. A Gradient Method for Constrained Extrema.

It is natural to inquire (as Samuelson has done in the linear case; see [4], pp. 17-22, 74-78) whether or not some variant of the gradient process could be used in problems where maximization may be subject to constraints. We shall consider the case where

---

1. Theorem 2 of [3] treats the case of $\Psi(u, v)$ convex in y.

2. Extension to the case where U and V are more general is immediate.

3. That is, Ψ is locally strictly concave in $u^{(2)}$.

4. Condition (b) of the above theorem is not needed for Uzawa's results.

(1) is replaced by,

(7) $f(\hat{x}) \geq f(x)$ for all x in X such that,

(7.1) $g_j(x) \geq 0$, $j \in M \equiv \{1, 2, \ldots, m\}$,

(7.2) $x_{i'} \geq 0$, $i' \in N'$, $N' \subseteq N = \{1, 2, \ldots, n\}$,

where M and/or N' may be empty, so that the case of an unconstrained maximum is covered. The case of equality constraints is also included, since

$$h(x) = 0,$$

is equivalent to the pair of inequalities,

$$h(x) \geq 0, \quad -h(x) \geq 0.$$

In view of the applicability of gradient processes to saddle-points, a method of deriving a gradient process for a constrained maximum is suggested by the Kuhn-Tucker [5] results which establishes the equivalence of the two kinds of problems when f and all the g_j's are concave and subject to a regularity condition (the "Constraint Qualification") on the g_j's which we shall tacitly assume in what follows. The Kuhn-Tucker theorem states that, provided the functions f and g_j are concave,[1] the Lagrangian expression,

$$\phi(u, v) \equiv f(x) + y'g(x),$$

has a saddle-point x at $(\bar{x}, \bar{y})$ over,

$$X = \{(x_1, \ldots, x_n): x_{i'} \geq 0, i' \in N'\},$$

$$Y = \{(y_1, \ldots, y_m): y_j \geq 0, j \in M\},$$

with $\bar{x} = \hat{x}$, where $\hat{x}$ maximizes $f(x)$ subject to (7.1) and (7.2).

In turn, one may apply the gradient process to the Lagrangian expression $\phi(x, y)$ and, in the light of the preceding section, convergence in the

1. Note that when an equality constraint $h(x) = 0$ is represented by two inequalities, $h(x) \geq 0$ and $-h(x) \geq 0$, the concavity requirement for both $h(x)$ and $-h(x)$ means that h must be linear.

x-components will occur provided that $\phi(x, y)$ is strictly concave in x for each y. But if $\phi(x, y)$ is to be strictly concave in x for each y, it is not sufficient that f and the g_j's be merely concave in x. On the other hand, it is sufficient if f is strictly concave while the g_j's are concave.

5. The Linear Case: Local Convergence

So far, we are in a position to treat by gradient processes (via conversion to saddle-points) any constrained maximization problem where the constraint functions g_j are all concave and the maximand f strictly concave. In particular, the g_j's might be linear. However, the requirement that f be strictly concave excludes the possibility of a linear maximand.[1] Hence, one is led to look for a device that would make it possible to handle a problem that is completely linear, or, more generally, where f and the g_j's are concave but not necessarily strictly concave.

A device of this type was explored in [6], where the Lagrangian expression, $\phi \equiv f + y'g$, was replaced by a modified Lagrangian expression,

$$\eta\phi(x, y) \equiv f + \sum_{j \in M} y_j (1 - h_j^{1 + \eta_j}),$$

$$h_j \equiv h_j(x) \equiv 1 - g_j(x),$$

$$\eta_j \geq 0.$$

It was shown in [6] that subject to certain regularity conditions $\eta\phi$ can be made locally strictly concave in the interior components of x by selecting sufficiently large values of the η_j's. (This statement, in fact, holds for any sufficiently regular f and g_j.) In combination with the preceding results, it follows that the gradient process applied to

---

1. Samuelson, as noted above, applied the gradient process to the purely linear case; he obtained constant amplitude oscillations rather than convergence.

the modified Lagrangian expression converges to the constrained maximum if the initial position is sufficiently close.

6. The Linear Case: Global Convergence.

In 1955-6, the problem of convergence in the linear case was reconsidered under the stimulation of the computational applications of the modified Lagrangian gradient method carried on at RAND by T. Marschak in connection with linear programming problems.[1] The problem of chief interest was whether or not convergence in the large could be expected. This work[2] led to results showing that if the modified Lagrangian gradient method is applied to a linear programming problem, convergence in the large of the x-components is guaranteed, although it may be necessary to use a variant (to be described below) of the modified Lagrangian expression.

To state these results, we must first introduce a concept which fits between concavity and strict concavity, to be called partial strict concavity.

Definition. $h(x)$ is said to be partially strictly concave if, for any x^1, x^2 such that $h(x^1) \neq 0$, $h(x^2) = 0$, the function $H(\theta) = h\left[(1-\theta)\, x^1 + \theta\, x^2\right]$, $0 \leq \theta \leq 1$, is a concave but not linear function of θ.

To simplify the exposition, consider the case where a unique maximizing point $\hat{x}$ is known to exist. Then it has been shown that,

Theorem A. Each x_i will converge to $\hat{x}_i$ under the gradient process applied to the unmodified Lagrangian expression ϕ provided f is concave,

---

1. See Appendix.

2. Carried out by the authors under the auspices of the Office of Naval Research and the Center for Advanced Study in the Behavioral Sciences, respectively.

each g_j is partially strictly concave in x, and all functions are analytic.

But this theorem, by itself, is inadequate to deal with the case where f and all the g_j's are linear. Thus we must fall back on a modified Lagrangian expression.

Definition. The function,

$$\rho(x, y) \equiv f(x) + \sum_{j \in M} y_j \rho_j [g_j(x)],$$

is called the ρ-modified Lagrangian expression for the maximization of f subject to $g_j(x) \geq 0$, $j \in M$ if the functions ρ_j are strictly increasing and $\rho_j(0) = 0$.

We then have,

Theorem B. Each x_i will converge to $\hat{x}_i$ under the gradient process applied to the ρ-modified Lagrangian expression provided f and g_j are concave and analytic and each ρ_j is strictly concave.

In this case the constraints $\rho_j [g_j(x)] \geq 0$ are equivalent to the constraints $g_j(x) \geq 0$ but the former satisfy the conditions of Theorem A.

Theorem B is applicable to completely linear problems, since f and g_j are only required to be concave.

The modified Lagrangian expression in section 5 above corresponds to choosing,

$$\rho_j^{(1)}(w) = 1 - (1 - w)^{1 + \eta_j}, \quad \eta_j > 0,$$

which satisfies the requirements of Theorem B for $w < 1$. It cannot be guaranteed to insure convergence in a linear programming problem unless $g_j(x) < 1$ throughout the gradient process. On the other hand,

$$\rho_j^{(2)}(w) = 1 - e^{-\eta_j w}, \quad \eta_j > 0,$$

has the required properties for all y and can safely be used in the large, i.e., for any initial values, the gradient process applied to the ρ-modified Lagrangian expression for a linear (or any concave) programming problem will converge.

7. Remarks on Computations.

In T. Marschak's experimental computations, the modified Lagrangian expression was formed using $\rho^{(1)}$, but a check-column was carried to make sure that $g_j(x)$ remained less than 1 during the process.

A check of the experimental computations was also offered by verifying that the "distance function,"

$$D = \sum_i (x_i - \bar{x}_i)^2 + \sum_j (y_j - \bar{y}_j)^2,$$

always decreased with time. (The theorems on convergence are based on the fact that $dD/dt < 0$ in a gradient process.) This check is only of interest in the experimental computations, since it requires knowledge of the unknown of the problem, viz., $\bar{x}$. For the computations themselves, see the Appendix.

References

[1] Arrow, K. J., and Hurwicz, Leonid, "The Gradient Method for Constrained Maxima," RAND P-928, 28 August 1956.

[2] Samuelson, Paul A., Foundations of Economic Analysis, Cambridge, Massachusetts: Harvard University Press, 1948.

[3] Arrow, K. J., and Hurwicz, Leonid, "A Gradient Method for Approximating Saddle Points and Constrained Maxima," RAND P-223, 13 June 1951.

[4] Samuelson, Paul A., "Market Mechanisms and Maximization," RAND P-69, 28 March 1949.

[5] Kuhn, H. W., and Tucker, A. W., "Nonlinear Programming," in J. Neyman (ed.), Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, Berkeley and Los Angeles: University of California Press, 1952, pp. 481-492.

[6] Arrow, K. J., and Hurwicz, Leonid, "Reduction of Constrained Maxima to Saddle-Point Problems," RAND P-706, 7 July 1955; also in J. Neyman (ed.), Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Berkeley and Los Angeles: University of California Press, 1956, Volume V.

APPENDIX

by

T. Marschak

This appendix reports on some computations in which the gradient method was applied to the ρ-modified Lagrangian expression of section 6 - with the functions ρ_j taking the form $\rho^{(1)}$ - for two linear programming problems. In the computations the η_j were taken to be the same for all j, i.e., $\eta_j = \eta$, $j = 1,\ldots,m$.

The aims of the computations were (1) to explore the major difficulties encountered in programming the gradient method (as applied to constrained maximizations problems) for a digital computer;[1] (2) to obtain some preliminary evidence as to the efficiency of the method in solving linear programming problems of different sizes as well as the effect of varying η and varying the Euclidean distance of the initial point from the equilibrium point.

(1) Major programming difficulties.

Two major programming problems were encountered: (a) the modification of a stable method for approximating differential equation systems so as to allow for "corner" conditions; (b) satisfaction of the constraint on the range of numbers which may occur.

(a) Modifying a stable approximating method.

If a digital computer is to be used in obtaining the paths of variables, as determined by a system of differential equations and

---

1. The RAND Johnniac was the machine used; the coding and machine work were done by Marvin Shapiro.

an initial point, the system must first be approximated by a system of difference equations. The simplest approximation to a system $d Z_{i/dt} = F_i (Z_1,\ldots,Z_r)$, $i = 1,\ldots,r$, is, of course, the difference-equation system

$$Z_i (t) = Z_i (t - 1) + h F_i [Z_1 (t - 1),\ldots, Z_r (t -1)]$$

$$\equiv Z_i (t - 1) + h F_i (t - 1) \qquad (h > o)$$

This system defines for each variable a series of connected straight-line segments approximating the true path of the variable (as defined by the differential-equation system). Unless h is taken sufficiently small, however, there is a danger that when the true path of a variable Z_i undergoes a sharp change in slope between t and t + 1, the approximating path between t + 1 and t + 2 (being based on the slope of the approximating path at t) will be an extrapolation of the true path as it was prior to the sharp change in slope. The approximating path may then deviate further and further from the true one. To exclude this danger may require h to be so small that convergence is uselessly slow.

Chiefly for this reason, more stable approximation methods have been devised. One of the most widely used is the Runge-Kutta method ([A], pp. 301-2) which was used in the experimental computations. In this method the iteration leading from $Z_i (t-1)$ to $Z_i (t)$ consists of the following steps:

(1) $Z_i^1 (t) = h F_i [Z_1 (t - 1),\ldots, Z_n (t - 1)]$

(2) $Z_i^2 (t) = h F_i [Z_1 (t - 1) + 1/2 Z_1^1 (t),\ldots, Z_n (t - 1) + 1/2 Z_n^1 (t - 1)]$

(3) $Z_i^3 (t) = h F_i [Z_1 (t - 1) + 1/2 Z_1^2 (t),\ldots, Z_n (t - 1) + 1/2 Z_n^2 (t)]$

(4) $z_i^4(t) = h F_i\left[z_1(t-1) + z_1^3(t), \ldots, z_n(t-1) + z_n^3(t)\right]$

(5) $z_i(t) = 1/6\left[z_i^1(t) + 2z_i^2(t) + 2z_i^3(t) + z_i^4(t)\right] + z_i(t-1)$.

This method has a low "inherent error," of the order of h^5 (See [A], p. 302).

Now, in our case F_i, in the differential-equation system, is of the form

$$F_i = \begin{cases} F_i^* \text{ if the ith argument is } > 0 \\ \max\ (F_i^*, 0) \text{ if the ith argument is } = 0. \end{cases}$$

No variable, is allowed, in other words, to become negative.

In formulating the Runge-Kutta approximation so as to give effect to this condition, it is clear that we cannot leave steps (1) to (5) as they are. For if F_i^* takes a large (in absolute value) negative value for the arguments shown on the right-hand side of an equation defining one of the above steps (i.e., if F_i takes such a value and if the ith argument is positive), then even for small h, $x_i(t)$ may be negative. To solve the difficulty several possibilities suggest themselves:

(i) We can impose the condition that $z_i(t)$ be the maximum of zero and the weighted average just given (in step (5)).

(ii) We can rewrite steps (1)-(4), everywhere replacing F by F^* and can then impose the condition that $z_i^k(t)$ be the maximum of zero and the expression on the right-hand side of the equality defining step (k), $k = 1, \ldots, 4$.

(iii) When the value $z_i(t)$ obtained at the end of the iteration is negative, we can repeat the iteration (starting again with $z_i(t-1)$) for a smaller h. If the value of $z_i(t)$ then obtained is again negative we repeat the iteration with a still smaller h. This process continues until $z_i(t)$ falls in a small pre-selected interval $(-\varepsilon, \varepsilon)$, $\varepsilon > 0$. At

that point z_i (t) is set equal to zero. For h sufficiently small, z_i (t) must fall in the given interval.

In choosing between these alternatives we must rely on intuition about the relative protection they afford against major deviation from the true paths of the variables, and on actual experience in using the alternatives.

Alternative (i) was attempted and failed to work properly: the "distance function," given in section 6, fluctuated instead of decreasing monotonically. If alternative (i) is used there is a danger that the following may occur (and this may have caused the "distance" fluctuation): one of the z_i^k (t), say $z_i^{\bar{k}}$ (t), may, because the value of F_i on the right-hand side of the equation defining step ($\bar{k}$) is large (in absolute value) and negative, become large and negative. The discrepancy between $z_i^{\bar{k}}$ (t) and the corresponding true value of z_i is then unusually large (as compared to the discrepancy in an interval where neither true nor approximating values are near zero). Then averaging the z_i^k (t) (in step (5)) and taking the maximum of the result and zero may well not erase the large discrepancy, e.g., the true value corresponding to a zero z_i (t) may be large (and positive).

Alternative (ii) on the other hand is certainly risky, since the true functions F_i are not used in the approximation. Thus unless h is small enough, there is a danger that, after taking a positive value, z_i in the true path descends sharply (with a large negative slope F_i^*) towards zero, but without reaching zero; while in the approximating path z_i is suddenly given the value zero.

Alternative (iii) appears to be the best of the three, according as it does with the general intuitive notion that the approximating "grid"

needs to be "finer" at points of discontinuity in the derivative of the true path. This alternative has been used and has worked quite well. Specifically, in the routine used, h is successively cut in half until $\beta_i(t)$ falls in the interval $(-\varepsilon, \varepsilon)$. ε was chosen to be just slightly greater than the average round-off error expected on the machine.

(b) Meeting the constraint on the permissible range of values.

On the RAND Johnniac, using a floating-point code (as was done in these computations), only numbers between 10^{-50} and 10^{50} can be dealt with.[1] Hence not all sets of values of h and η and of initial values of the variables can be used. Thus for a given η, the initial values and the values of h have to be chosen with some care. Though a large η may lead to rapid convergence for any given h, the largest value of h consistent with the range constraint for such an η may be small enough so that the actual time required for convergence is longer than for some smaller η.

2. The linear programming problems to which the gradient method was applied.

In a linear programming problem of the form:

Find x such that $\sum_{i \in \mathcal{N}} c_i x_i = \max$

subject to $x_i \geq 0$, $\sum_{i \in \mathcal{N}} a_{ij} x_i \geq b_j$, $j \in \mathcal{M}$

the ρ-modified Lagrangian expression of section 6 - with the functions ρ_j taking the form $\rho^{(1)}$ and all $\eta_j = \eta$ - is

$$\eta^{\rho}(x, y) = \sum_{i \in \mathcal{N}} c_i x_i + \sum_{j \in \mathcal{M}} y_j \left[1 - \left(\sum_{i \in \mathcal{N}} a_{ij} x_i - b_j + 1\right)^{1+\eta}\right]$$

The gradient process, which converges to the required solution (for any initial values of x_i), is given by the differential-equation system

---

1. On the I.B.M. 704, the permissible range is smaller.

$$dx_i/dt = \partial_\eta \mathcal{Q}/\partial x_i = c_i - (1+\eta) \sum_{j \in \mathcal{M}} a_{ij} y_j \left(\sum_{i \in \mathcal{N}} a_{ij} x_i - b_j + 1 \right)^\eta$$

if $x_i > 0$

maximum of this and zero if $x_i = 0$

$$dy_j/dt = -\partial_\eta \mathcal{Q}/\partial y_j = \sum_{i \in \mathcal{N}} (a_{ij} x_i - b_j + 1)^{1+\eta} - 1 \quad \text{if } y_j > 0$$

maximum of this and zero if $y_j = 0$

(a) <u>A three-variable three-inequality problem</u>

The first problem treated was as follows:

Find x_1, x_2, x_3 (≥ 0) such that

$\pi = x_1 + 2x_2 + 3x_3 = \max$ subject to

$$x_1 + x_2 + x_3 \leq 1$$
$$-x_1 - x_2 + x_3 \leq 0$$
$$-x_1 + 2x_2 \leq 0$$

The solution is $x_1^\circ = 1/3$, $x_2^\circ = 1/6$, $x_3^\circ = 1/2$, for which $\pi = 13/6 = 2.167^-$.

In the first computation η was taken to be 2, for which the equilibrium values of the y_j are

$y_1^\circ = 13/18 = .722^+$, $y_2^\circ = 5/18 = .278^-$, $y_3^\circ = 1/9 = .111^+$

The initial values chosen were 1 for x_1, 5 for x_2, 3 for x_3, 3 for y_1, 2 for y_2, and 3 for y_3.

In the second computation, η was taken equal to 7; for this case, the equilibrium values of the y_j are

$y_1^\circ = 13/48 = .271^+$, $y_2^\circ = 5/48 = .104^+$, $y_3^\circ = 1/24 = .0415$.

In the computation for the case $\eta = 2$ the initial value for h was

.0150. In the computation for the case $\eta = 7$, use of the same h would have been impossible, because very large numbers would have resulted in the expressions involving the exponent $\eta + 1$, while in other expressions there would have occured numbers so much smaller as to violate the restriction on range. The initial h used in the case $\eta = 7$ was .01. The initial values of h used in the two cases were approximately as large as they could be without violating the range constraint; from a practical point of view (i.e. if fast convergence is the aim) they are the proper values to choose. Nevertheless, they are sufficiently different so that the comparison of the two cases with respect to the effect of varying η is difficult. A floating-point code was used.

The following table partially summarizes (for selected iterations) the results of the computations for the three-equation problem, for the cases $\eta = 2$ and $\eta = 7$. For each iteration in the table are shown (to three decimal places) the current value of the x's, the y's, h (which determines the "size" of the approximating step), w and the "distance" $D = \sum_{i=1}^{3} \left[(x_i - x_i^0)^2 + (y_i - y_i^0)^2\right]$. In addition to the variables shown in the table, there were printed out (at every tenth iteration) the values of all derivatives and the values of the left-hand sides of the three inequalities. Printing out of the latter values made it possible to keep a running check as to violation of the requirement $g_j(x) \leq 1$ which, as indicated in section 6, must be met if the conditions of Theorem 3 are to guarantee convergence. In none of the computations—either for the 3-equation or the 5-equation problem—was this requirement not met.

We see from the table that only at some iteration between 330 and 340 did the computation for $\eta = 7$ "overtake" the computation for $\eta = 2$ - in the

Iteration number	h	x_1	x_2
Case n = 2			
0		0.000	1.000
10	$(19)(10^{-7})$	.291	0.000
20	.0156	.520	.000
70	.0156	.576	.000
150	.0156	.520	.000
300	.0156	.337	.137
330	.0156	.311	.171
340	.0156	.318	.165
700	.0156	.333327	.166662
Case n = 7			
0		0.000	1.000
10	10^{-8}	.092	1.077
20	10^{-9}	.003	.987
70	.01	.501	.044
150	.01	.498	.051
300	.01	.445	.069
330	.01	.412	.08
340	.01	.405	.091
64	.01	.33333334	.16666663

x_3	x_1	y_2	y_3	D	W
5.000	1.600	2.000	3.000	33.470	17.000
2.123	2.992	2.237	3.117	24.918	6.661
.284	3.090	2.405	3.065	23.458	1.373
.096	2.531	1.723	2.342	13.844	.864
.241	1.739	.767	1.206	4.106	1.242
.471	.865	.279	.110	.166	2.023
.483	.806	.274	.109	.088	2.102
.486	.791	.275	.120	.072	2.108
.499990	.722227	.277775	.111110	.000042	2.16662

5.000	1.600	2.000	3.000	33.470	17.000
5.008	1.614	2.010	3.001	35.558	17.272
4.867	1.661	2.017	3.001	34.354	16.579
.221	2.025	2.189	2.866	15.701	1.251
.257	1.316	1.417	2.073	6.603	.708
.448	.316	.126	.602	.350	1.927
.498	.460	.298	.317	.092	2.066
.498	.453	.287	.226	.047	2.080
.5000...	.722223	.27777...	.11111...	$(137)(10^{-10})$	2.16666...

sense of exhibiting a smaller Euclidean distance from the equilibrium point. After this iteration the distance in the computation for $\eta = 7$ decreases considerably more rapidly (with respect to number of iterations) than the distance in the computation for $\eta = 2$. For example, (this is not shown in the table) the distance becomes less than .01 at iteration 380 for $\eta = 7$, but not until iteration 550 for $\eta = 2$. In both computations it is necessary to cut the size of h at some iteration, primarily, as it happens, at early iterations. Each iteration averaged about four seconds for $\eta = 2$ (allowing for print-out time at every tenth iteration) and about five seconds for $\eta = 7$.

(b) A nine-variable, five-inequality problem

The second problem treated was to maximize $z = x_1 + x_3 + 5/4\ x_4 + 2x_5 + 2x_7 + 5/4\ x_8 - x_9$ subject to $x_i \geq 0$, $i = 1,\ldots,9$ and

(1) $x_9 \geq 1$

(2) $x_1 + x_2 - 1/2\ x_9 \geq 0$

(3) $x_5 + x_6 - 1/4\ x_9 \geq$

(4) $x_3 + x_4 - 1/2\ x_2 - 3/8\ x_6 \geq$

(5) $x_7 + x_8 - 1/3\ x_2 - 5/8\ x_6 \geq$

The solution is $x_1^\circ = 0$, $x_2^\circ = 1/2$, $x_3^\circ = 0$, $x_4^\circ = 1/4$, $x_5^\circ = 1/4$, $x_6^\circ =$, $x_7^\circ = 1/6$, $x_8^\circ = 0$, $x_9^\circ = 1$, $z = 7/48$.

In the two computations which were performed η was taken = 2 so that at equilibrium $y_1 = 7/144$, $y_2 = 31/72$, $y_3 = 2/3$, $y_4 = 5/12$, $y_5 = 2/3$.

In the first computation the initial point chosen was that shown (iteration 0) in the first part of the following table, for which the

Iteration number	h			x_1	x_2	x_3
Close initial point						
[illegible]		.327	.25[illegible]	[illegible]	.4	[illegible]
1[illegible]	.01	.3[illegible]9	.2[illegible]8	[illegible]	.44[illegible]	[illegible]
4[illegible]	.[illegible]01	.216	.0[illegible]	[illegible]	.44[illegible]	[illegible]
13[illegible]	.[illegible]01	.[illegible]13	.0[illegible]	[illegible]	.439	[illegible]
25[illegible]	.1	.[illegible]8[illegible]	.4[illegible]1	[illegible]	.411	[illegible]
2[illegible]0	.[illegible]1	.[illegible]4	.1463[illegible]	[illegible]	.5[illegible]	[illegible]
Distant initial point						
[illegible]		143.1[illegible]	29.5[illegible]	5.[illegible]	[illegible].[illegible]	[illegible].[illegible]
1[illegible]	.[illegible]001	141.53[illegible]	[illegible]6.545	4.29[illegible]	[illegible].489	1.78[illegible]
13[illegible]	1.[illegible]-6	1[illegible]7.541	11.[illegible]76	1.798	1.319	.[illegible]38
8[illegible]	.1	8[illegible].[illegible]7	.059	[illegible].[illegible]0	[illegible].[illegible]	[illegible].[illegible]0
1[illegible]5	.1	[illegible]5.429	.8[illegible]5	[illegible].[illegible]0	.188	[illegible].[illegible]
17[illegible]	.[illegible]1	1.851	.913	[illegible].[illegible]0	[illegible].[illegible]306	[illegible].[illegible]0
[illegible]	.[illegible]1	[illegible].86	.393	[illegible].[illegible]	.371	[illegible].[illegible]0

x_4	x_5	x_6	x_7	x_8	x_9	y_1	y_2	y_3	y_4	y_5
.[illegible]	.[illegible]	[illegible]	.3	[illegible]	1.[illegible]	.[illegible]8	.625	1.[illegible]	.6	1.[illegible]
.160	.142	.[illegible]21	.152	[illegible]	1.11[illegible]	.[illegible]83	.603	.981	.589	1.[illegible]2
.125	.1[illegible]6	.[illegible]01	.[illegible]18	[illegible]	1.1[illegible]1	.1[illegible]4	.55[illegible]	.88[illegible]	.516	.950
.125	.1[illegible]7	.[illegible]	.[illegible]16	[illegible]	1.015	.1[illegible]4	.549	.878	.513	.947
.15[illegible]	.151	.[illegible]**	.[illegible]21	[illegible]	.95[illegible]	.[illegible]85	.47[illegible]	.753	.427	.81[illegible]
.[illegible]5	.[illegible]5	[illegible]	.166	[illegible]	1.[illegible]	.[illegible]49	.431	.667	.418	.667

4.[illegible]	4.[illegible]	1.[illegible]	5.[illegible]	[illegible]	3.000	4.000	5.000	4.000	3.000	2.000
3.737	3.695	.935	4.753	1.753	3.31[illegible]	4.031	5.312	4.125	3.127	2.243
[illegible].092	1.884	.946	3.009	[illegible].[illegible]	3.321	4.489	5.911	4.642	3.678	3.229
.[illegible]	.[illegible]36	.[illegible]	.[illegible]	[illegible].[illegible]	.[illegible]06	4.495	5.553	4.613	3.792	3.592
.[illegible]	[illegible].[illegible]	.[illegible]59	[illegible].[illegible]	[illegible].[illegible]	.[illegible]805	1.745	2.987	3.015	2.847	2.740
.[illegible]	[illegible].[illegible]	.[illegible]64	[illegible].[illegible]	[illegible].[illegible]	[illegible].913	[illegible].286	0.971	1.272	1.150	1.297
.[illegible]5	.171	[illegible].[illegible]	0.048	[illegible].[illegible]	[illegible].913	[illegible].000	0.401	[illegible].665	[illegible].435	0.732

distance from equilibrium is .327+. In the second computation the initial point was that shown in the second part of the table (iteration 0) for which the distance from equilibrium is 143.1. The initial value of h was taken to be .01 in both computations. Both computations are summarized, for selected iterations, in the table.

It is seen that at any iteration up to about the 2000th, there has been achieved in the distant-point case a much greater absolute decrease in distance from the equilibrium point than in the near-point case, though roughly the same relative decrease in distance has been achieved. In both computations it is necessary at some iteration to cut the size of h. It is worth noting that it is not true that once a variable attains the value zero, it stays at zero if zero is its value at equilibrium. The variable x_6, in the distant-point computation, provides the counter-example.

Reference

[A] Scarborough, J. B., Numerical Mathematical Analysis, Baltimore, Maryland: Johns Hopkins Press, 1955, p. 157.